

*MBL200*

# *mobile PC & devices*

## Overview: Designing Mobile PC Applications for “Longhorn”

Ravi Soin

PM, Windows Client Mobility

Microsoft Corporation

[ravisoin@microsoft.com](mailto:ravisoin@microsoft.com)

**PDC**<sup>03</sup>

Make the connection

**Microsoft**<sup>®</sup>

# Agenda

- Mobile Platform
- Why Mobile PC's?
- Longhorn Mobility Focus Areas
  - Demos & Code Sample
- Call To Action
- Resources
- Q&A

# Microsoft Mobility Platform

## Smart Personal Objects

- One-way network
- Information consumption



## Smartphone

- Integrated phone with PDA
- Data viewing
- Interoperability with Exchange
- .NET Compact framework



## Pocket PC Phone

- Integrated PDA with phone
- View and some data entry
- Interoperability with Office, Exchange and SQL Server



## Notebook PC

- Complex document authoring, editing and reading
- Keyboard centric at the desk
- Keyboard and mouse input methods
- Full .NET framework available



## Tablet PC

- Complex document authoring, editing and active reading
- Note taking and ink annotating
- Keyboard centric at the desk, pen & keyboard away from the desk
- Keyboard, mouse plus pen, ink, and speech input methods
- Full .NET framework preinstalled
- Handwriting and speech recognition

Windows  
CE+

Increased

Full .NET framework available  
Longhorn API's

# WinFX Developer Preview Longhorn Architecture

## Tools

Visual Studio .net  
Visual Basic .net  
Visual C++ .net  
Visual C# .net  
Visual J# .net

### Client Application Model

#### Avalon

System.Windows

#### Windows Forms

System.Windows.Forms

### Web & Service Application Model

#### ASP.NET / Indigo

System.Web

### Data Systems Application Model

#### Win FS

System.Storage

#### Yukon

System.Data.SqlServer

### Mobile PC & Devices Application Model

#### Mobile PC Optimized

System.Windows  
System.Windows.Forms

### Command Line

System.Console

#### NT Service

System.ServiceProcess

## Presentation

### Avalon

Document UI Media

Desktop Services Desktop Window Manager Controls Interop Engine

Presentation Object Manager Desktop Composition Engine Application Services

#### Media Services

Animation and Composition Media Processing Capture and Sourcing Designer Services

Hardware Rendering Software Rendering and Sinks Controls

### Windows Forms

### ASP.NET

Active UI Engine

Page/Script Composition

Personalization and Profiling Services

Membership and Security Services

## Data

### WinFS

Object T/SQL XML

Services Synchronization (Windows, Windows 32..)

Schema People Calendar Media Document

Data Model Items Relationships Extensions

File System Services (Metadata Handlers..)

### ADO.NET

Object Spaces

Data Set

SQL XML

Providers

## Communication

### Indigo

Service

Channels (Datagram, Reliable, Peer, ...)

Policy Engine Channel Security Message Encoder

Transport Channels (IPC, HTTP, TCP...)

Communications Manager (Port)

Messaging Services

Queuing Eventing Routing

System Services

Transaction Federation

### Collaboration

People and Groups

Collaboration History

Real-Time Activities

Signaling

## Base Operating System Services

### CLR

Base Class Libraries

Application Deployment Engine (Click-Once)

Memory Manager

Code Execution

Loader

Security

Serialization

Hosting Layer

GDI/GDI+ Window Manager Global Audio Engine DirectX Graphics Graphics Drivers

DDI Input Manager Audio Drivers DirectX Graphics Mini port

Plug and Play Memory Manager Power Manager Config Manager Process Manager Security Reference Monitor LPC Facility

Kernel

Hardware Abstraction Layer

Transaction Lightweight Transaction Coordinator Kernel Transaction Logging Service

Identity Security System

Cache Manager

IO Manager

SCSI/FC

### Storage

Backup / Restore

Redirection

Transacted NTFS

Universal Data Format

FAT 16/32

Service

Distributed File System

File System Manager

Cache Manager

IO Manager

SCSI/FC

Virtual Disk Management

Cache Manager

IO Manager

SCSI/FC

IO Manager

SCSI/FC

Cache Manager

IO Manager

SCSI/FC

IO Manager

SCSI/FC

IO Manager

SCSI/FC

Network Class Library

Network Services Demand Activation and Protocol Health

PNRP Native WiFi SIP TCP Listener UDP Listener IPC Listener

Internet Connection Firewall

Protocols Filter Engine TPC, UDP, IPV4, IPV6 IPSEC QOS HTTP Listener

Device Drivers 802.3 802.11 ..

Models

Framework

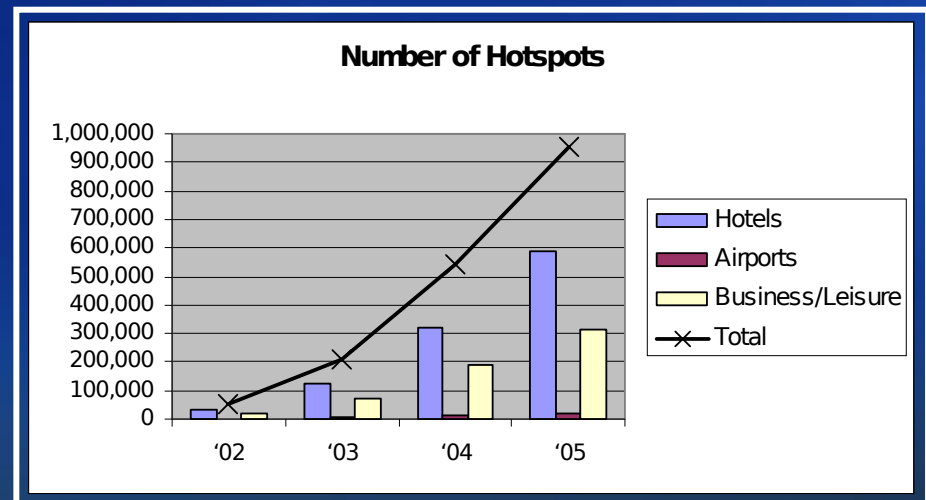
Kernel Mode

# Mobile PC Opportunity

- Faster than desktop growth in key markets
- Faster rate of innovation than desktops
- Mobile ecosystem firming up
  - Customer demand for real-time information
  - Wireless hotspots
  - WWAN access
- Segments
  - Corridor warriors
  - Road warriors

		Unit Sales		Pct of Market		Growth
		FY01	FY02	FY01	FY02	FY02 over FY01
US	Desktop	38.65	34.84	80.00%	77.60%	-9.87%
	Portable	9.66	10.07	20.00%	22.40%	4.32%
	Total	48.31	44.91			-7.03%
Europe	Desktop	25.15	22.47	78.70%	75.00%	-10.67%
	Portable	6.8	7.48	21.30%	25.00%	9.98%
	Total	31.95	29.95			-6.28%
Japan	Desktop	7.89	6.23	53.00%	50.20%	-20.97%
	Portable	6.99	6.18	47.00%	49.80%	-11.68%
	Total	14.88	12.41			-16.61%

Source: IDC PC Tracker (June 2002)  
Sales in millions of Units





# Longhorn Mobility Focus Areas

- The PC is **ready**; Users do not think actively about power management during an essential task
- The PC is **connected**; They have the ability to connect their PC to the best possible network to get to their business critical information and LOB apps
- The data is **available** to the user; They have instant (online and offline) access to their critical applications and data when they need it
- The PC is **usable**; Users are comfortable using their mobile PC through varying ergonomic conditions and activities

# Ready - Power Management

## Scenario Overview

- Whether in a presentation, taking notes in a meeting or checking email wirelessly, the mobile user is able to complete the tasks with minimal interruption and is aware of battery status

## Features

- Applications notified via power APIs to retrieve power status and scale accordingly
- Increased battery life on portable computers will allow users to be more productive while mobile
- PDA-like system availability and responsiveness means little to no wait time for users as they transition from standby
- Simplified power schemes and improved user experience

# Power Tile and Power Aware Avalon Sample

# demo

**PDC**<sup>03</sup>

Make the connection



# Power Aware Avalon Sample

```
private object UpdateForPowerStatus (object arg)
{
    if
    (WinForms.SystemInformation.PowerStatus.ACLineStatus == WinForms.ACLineStatus.Offline)
    {
        HidePreview (null, null);
        _animBase = 25;
        mainVideo.Stretch = Stretch.None;
    }
    else
    {
        _animBase = 250;
        mainVideo.Stretch = Stretch.Fill;
    }
}
```

# Windows Power Management

## Impact on applications

- Improperly designed applications may:
  - ▣ Prevent system power state transitions
  - ▣ Cause excessive power consumption
    - ▣ Poor battery life
    - ▣ Increased thermal load
- Many failure cases:
  - ▣ Application vetoes system sleep request, presents UI
    - ▣ *When the screen is off or laptop lid is closed!*
  - ▣ Application fails device eject query
    - ▣ *User can not un-dock the laptop!*
  - ▣ App renders nifty graphics, using lots of energy
    - ▣ *Even after the screen is powered off!*

# Windows Power Management

## Impact on applications

- Many failure cases:
  - System wakes, network connection or data not available
    - Application hangs!
  - System wakes, device is not available
    - Application hangs!
  - Application calls power apis to prevent system standby, screen blanking – neglects to call api to release
    - Machine never sleeps, battery drains, user's policy choices aren't respected...

# Intelligent Application Power Considerations

- P0
  - Proper handling of sleep and shutdown requests
  - Efficient use of CPU, memory, hard drive and other devices
    - Avoid polling
    - Avoid dynamically increasing timer frequency
    - Reduce or suspend background processing
- P1
  - Appropriate overriding of idle timers
  - Awareness of device power states
- P2
  - Scale app functionality to extend battery life e.g.
    - UI effects (transparency, “chrome”, transitions)
    - Periodic checking for product updates
    - Background content processing
- Visit talk: MBL304 to learn more details about Power Awareness

# Connected - Anywhere Wireless Networking

## Scenario Overview

- Users work **when** and **where** they want with smart, high-performance, secure wireless networking

## Features

- Longhorn PC automatically connects to fastest wired or wireless network based on user or IT pro preference (speed, security level, etc)
- Enhanced user experience through zero-config networks
- Support for Wireless WANs

## Benefits

- Applications notified of changes in the network
  - Network Location Awareness version 2
- Confidence that connection and data is secure
- Productivity is location independent
- Performance comparable to Ethernet
- Connect, collaborate & share information with others easily



# Characteristics of Mobile Nodes

- Availability
  - The default state for many mobile devices is in fact disconnected
- Delay
  - Mobile nodes on wireless network generally have greater delay. On WLAN networks it is typically 2-5 times greater; on WWAN networks 100-1000 times greater
- Loss
  - Packets are dropped more often
  - Some networks add delay to deal with loss
- Power Management
  - Effects the speed at which the network responds
  - Sleep cycles delays sends and receives

# Network Aware Guidelines

- Have intelligent error handling systems, which keep the user informed
- Do not hang when OS is accommodating network issues i.e.
  - Network problems when accessing documents in “My Network” causes hang
  - OS waking up network devices
- Detect network changes and react transparently
  - Treat the network as an unreliable resource
  - Limit the number of small transactions. Round trips between server and host are expensive
- Optimize for bandwidth
  - Use the best connection – NLA 2 API
- Test the applications in environments that have packet loss and low bandwidth
- Visit Talks MBL301 and MBL303 for details on Network Awareness

# Longhorn Collaboration While Mobile

## ...Making collaborative software easier to build

- Common way to address people in the system
  - Developers no longer need to build address-books as part of their application
- Platform provides services for identities
  - Developers can uniformly identify and authenticate users
- Common way to refer to past communications
  - Developers no longer need to build call logs as part of their application
- Visit talks:
  - CLI310 to learn more details about People & Groups Controls
  - WSV306 to learn more details about Indigo: Building Peer-to-Peer Applications

# Connected - Location Awareness

## Scenario Overview

- Longhorn can determine the computer's location based on input provided to the PC from a GPS receiver, wireless access point triangulation and user-entered information in order to adjust settings for firewall, peripherals or applications

## Features

- Location API set
- Aggregates all location information
- Supports an extensible location schema
- Supports an extensible plug-in model for new sources of location information

Visit talk: MBL 305 to learn more details about Location Awareness in Longhorn

# Available - PC And Device Synchronization

## Scenario Overview

- Mobile users can use a Longhorn PC to connect and synchronize content with PCs and devices to have the most up-to-date information

## Feature

- Hub for all synchronization between your PC and your applications, services and devices
- UI framework where you build a simple plug-in to leverage this common sync experience

## Benefits

- Consistent sync experience for applications
- Process Host for synchronization
- Common Status / Progress UI
- Conflict Store / Conflict UI
- Scheduling
  - Time and Event based



# Plugging In Your Application

## Writing a Sync Handler

- ISyncMgrSynchronize is the primary interface to communicate with handlers
- Key Methods:
  - Initialize
    - Tells handler which event is occurring (example: net connect etc.)
  - PrepareForSync
    - Tells the handler which items to sync in this session
  - Synchronize
    - Informs the handler to start synchronization
  - GetItemObject
    - Used to retrieve object interfaces (example: IExtractIcon)
  - ShowProperties
    - Used to launch handler specific configuration UI
  - SetItemStatus
    - Used to interrupt synchronization due to a user action

# Working With Data In Winfs

## Using WinFS Sync

- Data stored in WinFS should be synchronized using WinFS Sync
- WinFS Sync provides an adapter model to allow you to sync data from WinFS to:
  - Other WinFS Stores
  - Another store
  - A service
  - Etc.

By creating or using an existing sync adapter

- The UI for WinFS Sync is Synchronization Manager
- A sample synchronization handler for WinFS Sync is also provided in the PDC SDK
- Visit talk: CLI324 to learn more details about WinFS Sync

# Usable - Multiple Display Support

## Scenario Overview

- Patrick arrives to give a presentation. He connects to the projector, which is automatically detected and configured and he is able to give the presentation and take notes on his laptop at the same time
- Melissa undocks her laptop, goes to a meeting and when she returns, her existing display settings and placement of icons and windows are automatically restored on the external display

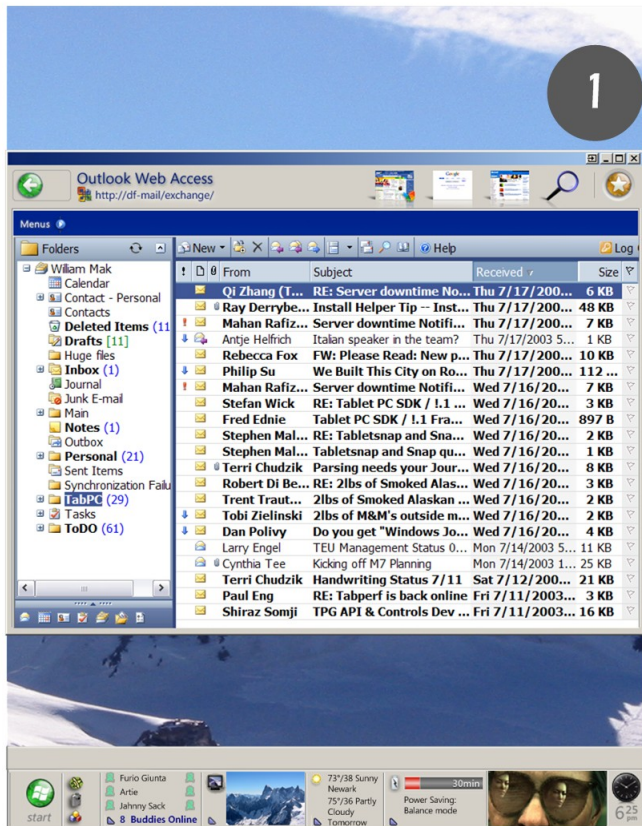
## Features

- Auto-detection of arrival & removal of external displays
- Automatic restoration of display settings and window placement
- Improved control of extended desktop
- Wireless presentation to UPnP projector
- Wireless presentation to multiple mobile PC's

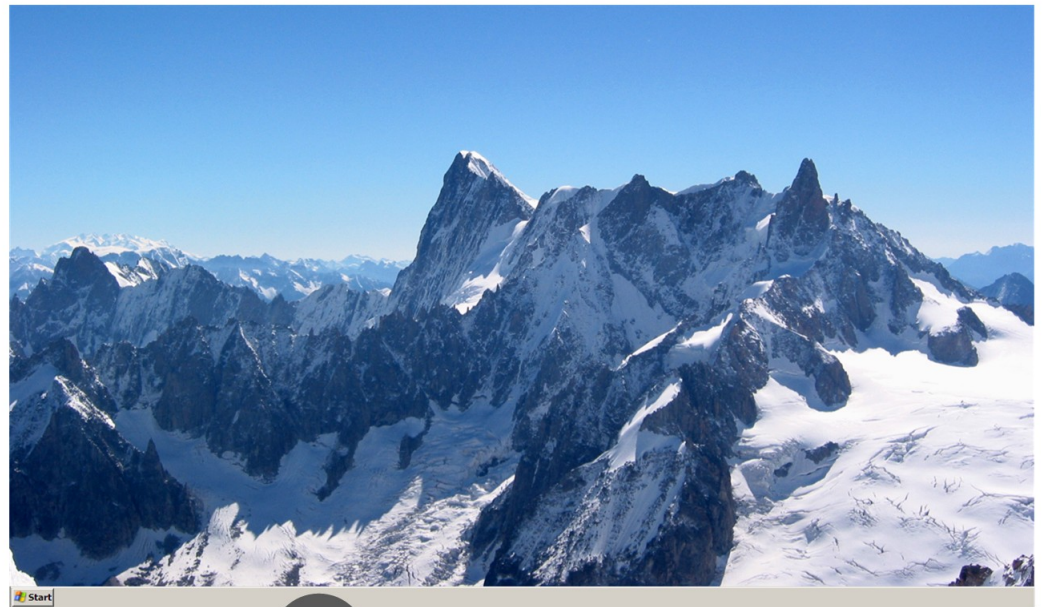
## User Benefits

- Enhanced PC mobility away from one's desk
- Increased productivity for meeting room scenarios
- Increased productivity at desk due to enhanced screen area and independent control of monitors

# Bump – Mockup (Draft)



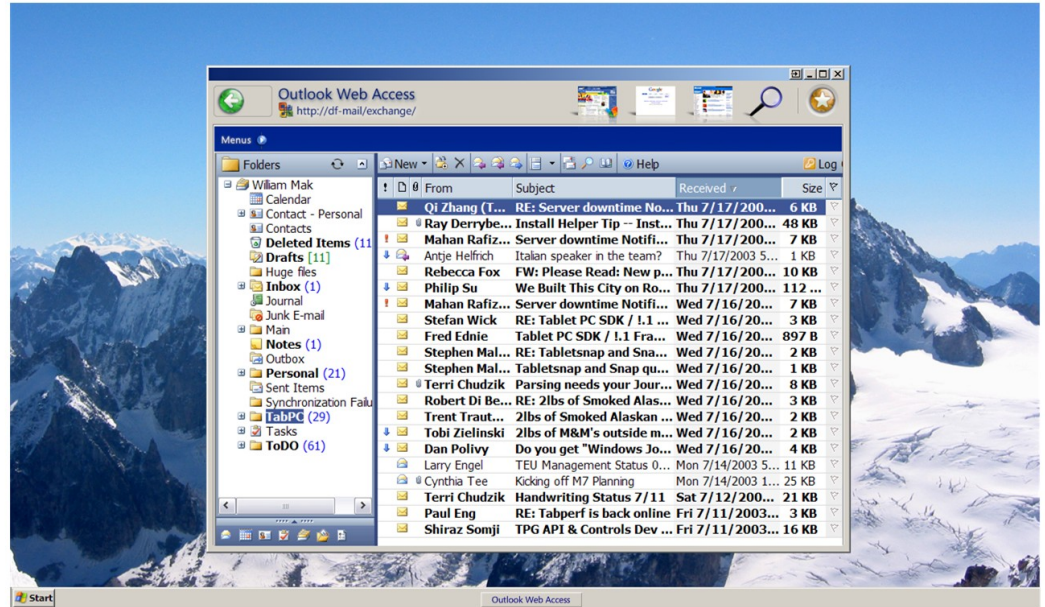
Display bar part



Bumping a window



# Bump – Mockup (Draft)



Window “bumped”



# Call To Action

- Keep mobility in mind when designing apps
- Focus on nailing the basics – these will impact your application behavior
  - ▣ Readiness - Power aware
  - ▣ Connected - Network aware
  - ▣ Available - Offline data/sync
  - ▣ Usable – Multimon support

# Related Talks

- Readiness
  - Fundamentals: MBL304
- Connected
  - Communication: MBL301, MBL303, MBL305, ARC382, ARC480, CLI310, WSV306
- Available
  - Data: MBL02, CLI320, CLI324
- Usable
  - Presentation: CLI350, CLI351
- Panels

# Community Resources

Get your questions answered!

- Newsgroups:

- microsoft.public.windows.developer.winfx.general

- Mobile PC & Devices Lounge: middle of the Exhibit Hall

- connect with Microsoft product teams, and PDC 2003 Speakers

- Ask The Experts:

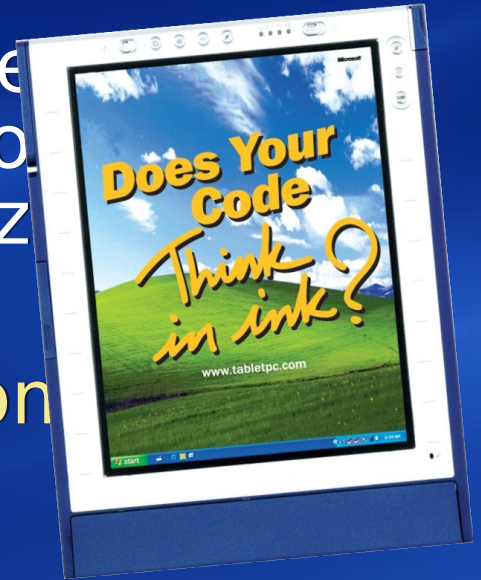
- Tuesday 7 pm – 9 pm in Hall G,H

- PDC Weblogs:

- <http://pdcbloggers.net>

# Calling All Developers! Win \$15,000 Cash

- Microsoft and PC Magazine invite you to develop the best PowerToys and win one of four fabulous prizes.
- Learn how you can win:  
[www.doesyourcodethinkinink.com](http://www.doesyourcodethinkinink.com)
- Get a free t-shirt and 64mb USB drive at the Tablet PC booth. Limited quantity available so don't miss this exciting opportunity.



# Questions And Answers



